

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

(11)特許出願公開番号

特開2003-152551

(P2003-152551A)

(43)公開日 平成15年5月23日(2003.5.23)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
H 0 3 M 13/27		H 0 3 M 13/27	5 J 0 6 5
13/23		13/23	5 K 0 1 4
13/29		13/29	5 K 0 2 2
H 0 4 B 1/707		H 0 4 L 1/00	F
H 0 4 L 1/00		H 0 4 J 13/00	D

審査請求 有 請求項の数18 O L (全 18 頁)

(21)出願番号 特願2001-353675(P2001-353675)

(22) 出願日 平成13年11月19日(2001. 11. 19)

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 發明者 丸 次夫

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100105511

弁理士 鈴木 康夫 (外1名)

Fターム(参考) 5J065 AA01 AA03 AB01 AC02 AD10

AG06 AH02 AH03 AH06 AH09

AH21

5K014 AA01 BA10 FA16 HA10

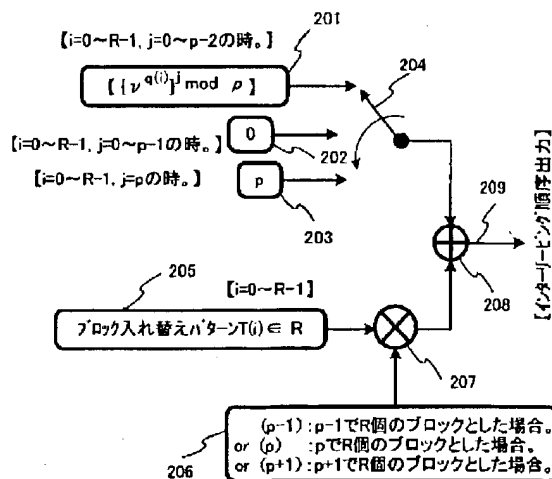
5K022 EE02 EE31

(54)【発明の名称】 インターリーピング順序発生器、インターリーバ、ターボエンコーダ、及びターボデコーダ

(57) 【要約】

【課題】 移動体通信システムに用いられるターボデコーダの内部インターリーブを、比較的少ないメモリ容量により実現する手段を提供する。

【解決手段】 データ長を、素数 p をベースにした長さ p で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成し、標数が前記素数 p の有限体の元を、原始元 α に対し前記整数の冪乗としてそれぞれ $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ のごとく生成し、これを前記有限体上で j 乗してそれぞれ $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を実時間で生成する手段201を備え、第0の順序入れ替えは、ブロック入れ替えパターン記録手段205からの出力を p 倍した値に1を順次足し合わせて行い、第 j の順序入れ替えは、前記手段205からの出力を p 倍した値に前記手段201で生成した値を実時間で順次足し合わせる操作を、 $j=1 \sim (p-2)$ において繰り返す。



【特許請求の範囲】

【請求項1】 データ長を、素数 p をベースにした長さ p で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成する手段と、標数が前記素数 p の有限体の元を、原始元 α に対し前記 $q_0, q_1, q_2, \dots, q_{R-1}$ の冪乗としてそれぞれ $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ のごとく生成し、記憶する手段と、

前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を前記有限体上で j 乗してそれぞれ $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を生成する手段と、前記ブロックの入れ替えを行う為の予め決められたブロック入れ替えパターンを生成または記録する手段と、第0の順序入れ替えにあたっては、前記ブロック入れ替えパターンを生成または記録する手段からの出力を p 倍した値に1を順次足し合わせて行い、第 j の順序入れ替えにあたっては、前記ブロック入れ替えパターンを生成または記録する手段からの出力を p 倍した値に前記生成した $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を順次足し合わせる操作を、 $j=1 \sim (p-2)$ において繰り返す手段と、を備えていることを特徴とするインターリービング順序発生器。

【請求項2】 前記 $j=1 \sim (p-2)$ の繰り返しの際して、前記生成及び記憶した $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を順次有限体における高速乗算器に入力することによって前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ の値を逐次的に更新する手段を有することを特徴とする請求項1に記載のインターリービング順序発生器。

【請求項3】 前記 $j=p-1$ において前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ に相当する値を全て0とする手段を有することを特徴とする請求項1または2に記載のインターリービング順序発生器。

【請求項4】 データ長を、素数 p をベースにした長さ $p-1$ で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成する手段と、標数が p の有限体の元を、原始元 α に対し前記 $q_0, q_1, q_2, \dots, q_{R-1}$ の冪乗としてそれぞれ $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ のごとく生成及び記憶する手段と、

前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を有限体上で j 乗してそれぞれ $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を生成する手段と、

前記ブロックの入れ替えを行う為の予め決められたブロック入れ替えパターンを生成または記録する手段と、第0の順序入れ替えにあたっては、前記ブロック入れ替えパターンを生成または記録する手段からの出力を $p-1$ 倍した値に1を順次足し合わせて行い、第 j の順序入

れ替えにあたっては、前記ブロック入れ替えパターンを生成し記憶する手段からの出力を $p-1$ 倍した値に前記生成した $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を順次足し合わせる操作を、 $j=1 \sim (p-2)$ において繰り返す手段と、を備えていることを特徴とするインターリービング順序発生器。

【請求項5】 前記順次足し合わせを行った値に対し1を減算する手段を有することを特徴とする請求項4に記載のインターリービング順序発生器。

10 【請求項6】 前記 $j=1 \sim (p-2)$ の繰り返しの際して、前記生成及び記憶した $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を順次有限体における高速乗算器に入力することによって前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ の値を逐次的に更新する手段を有することを特徴とする請求項4または5に記載のインターリービング順序発生器。

【請求項7】 データ長を、素数 p をベースにした長さ $p+1$ で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成する手段と、標数が前記素数 p の有限体の元を、原始元 α に対し前記 $q_0, q_1, q_2, \dots, q_{R-1}$ の冪乗としてそれぞれ $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ のごとく生成し、記憶する手段と、

前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を前記有限体上で j 乗してそれぞれ $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を生成する手段と、前記ブロックの入れ替えを行う為の予め決められたブロック入れ替えパターンを生成または記録する手段と、

30 第0の順序入れ替えにあたっては、前記ブロック入れ替えパターンを生成または記録する手段からの出力を $p+1$ 倍した値に1を順次足し合わせて行い、第 j の順序入れ替えにあたっては、前記ブロック入れ替えパターンを生成または記録する手段からの出力を p 倍した値に前記生成した $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を順次足し合わせる操作を、 $j=1 \sim (p-2)$ において繰り返す手段と、を備えていることを特徴とするインターリービング順序発生器。

40 【請求項8】 前記 $j=1 \sim (p-2)$ の繰り返しの際して、前記生成及び記憶した $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を順次有限体における高速乗算器に入力することによって前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ の値を逐次的に更新する手段を有することを特徴とする請求項7に記載のインターリービング順序発生器。

【請求項9】 $j=p-1$ において前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ に相当する値を全て0とする手段を有することを特徴とする請求項7または8に記載のインターリービング順序発生器。

50 【請求項10】 $j=p$ において前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ に相当する値を全て0とする手段を有することを特徴とする請求項7または8に記載のインターリービング順序発生器。

$q_1)^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ に相当する値を全て p とする手段を有することを特徴とする請求項7~9のいずれかに記載のインターリーブング順序発生器。

【請求項11】 前記インターリーブング順序発生器からの出力信号がインターリーブ対象範囲を超えた場合、該信号をスキップして次の該範囲内の信号を使用する手段を有することを特徴とする請求項1~10のいずれかに記載のインターリーブング順序発生器。

【請求項12】 前記有限体における高速乗算器を複数個備え、前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ の値の更新を、前記複数の有限体における高速乗算器で分担することにより、前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ の値の更新を、複数個同時に実行する手段を有していることを特徴とする請求項1~11のいずれかに記載のインターリーブング順序発生器。

【請求項13】 前記有限体における高速乗算器を2個備え、前記生成及び記憶した $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を、偶数乗数と奇数乗数に分割した $\alpha^{q_0}, \alpha^{q_2}, \alpha^{q_4}, \dots, \alpha^{q_{R-2}} \pmod{p}$ と $\alpha^{q_1}, \alpha^{q_3}, \alpha^{q_5}, \dots, \alpha^{q_{R-1}} \pmod{p}$ に対して、前記2個の高速乗算器を割り当て、前記有限体上で j 乗することにより、 $(\alpha^{q_0})^j, (\alpha^{q_2})^j, \dots, \alpha^{q_{R-2}} \pmod{p}$ と $(\alpha^{q_1})^j, (\alpha^{q_3})^j, \dots, \alpha^{q_{R-1}} \pmod{p}$ の値を、並行して同時に更新する手段を有していることを特徴とする請求項12に記載のインターリーブング順序発生器。

【請求項14】 請求項1~13のいずれかに記載のインターリーブング順序発生回路の出力を、データが蓄積されたメモリのアドレス信号とし、該アドレス信号により前記メモリからデータの読み出しを行うことにより前記データの順序入れ替えを行う手段を有していることを特徴とするインターリーブ。

【請求項15】 請求項1~13のいずれかに記載のインターリーブング順序発生回路の出力を、データを蓄積するメモリのアドレス信号とし、該アドレス信号により前記メモリにデータを書き込むことにより前記データの順序入れ替えを行うことを特徴とするインターリーブ。

【請求項16】 請求項14に記載のインターリーブを、ターボエンコーダの内部インターリーブとすることを特徴とするターボエンコーダ。

【請求項17】 請求項14または15に記載のインターリーブの内少なくとも一方をターボデコーダの内部インターリーブとし、他方を内部デインターリーブとすることを特徴とするターボデコーダ。

【請求項18】 請求項1~13のいずれかに記載のインターリーブング順序発生器の出力を、データが蓄積されたデュアルポートメモリの読み出し用アドレス信号としてデータ内容の読み出しを行い、予め決められた値で

遅延を施した該アドレス信号を書き込み用アドレス信号としてデータ内容の書き込みを行うことによりターボデコーダの内部インターリーブと内部デインターリーブを同時に実現したことを特徴とするターボデコーダ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、IMT2000（第三世代移動通信システム）のW-CDMAで用いられる素数インターリーブ（prime interleaver）に関し、特にインターリーブング順序発生器の為のメモリを削減したインターリーブング順序発生器、インターリーブ、ターボエンコーダ及びターボデコーダに関する。

【0002】

【従来の技術】広域DS-SS-CDMA（W-CDMA）は、第三世代の移動通信システム（IMT2000）無線アクセス方式（RAN）の一つとして標準化され、その中で素数インターリーブと呼ばれるターボコーディング用の内部インターリーブが規格化されている。その詳しい記述は「3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD) (Release 1999) 3G TS25.212 V3.3.0 (2000-06) 4.2.3.2.3章 Turbo code internal interleaver 16頁~20頁」に開示されている。

【0003】このターボエンコーダは、複数のコンポーネントエンコーダで構成されており、各コンポーネントエンコーダ間のパリティ系列の相関を薄くする為にインターリーブが用いられ、インターリーブを介して各コンポーネントエンコーダを接続する構成になっている。このインターリーブはターボコードの性能を発揮する上で重要な役割を果たしている。

【0004】図15は、従来のターボエンコーダの構成例を示す図である。同図に示す様に、ターボエンコーダは、複数の再帰的組織的畳み込みコンポーネントエンコーダ1502、1503と、インターリーブ1501を具備して構成されている。各再帰的組織的畳み込みエンコーダ1502、1503は加算器と単位遅延素子によって構成されており、ターボエンコーダに入力される情報系列1ビットに対して情報ビット、パリティビット1並びにパリティビット2の3ビットが出力される。パリティビット1とパリティビット2の相関を薄くする為、コンポーネントエンコーダ1503の前にインターリーブ1501を挿入している。

【0005】図16は、従来のターボデコーダの構成例を示す図である。ターボデコーダは、二つの軟入力軟出力デコーダ（soft in soft out decoder; 以下、SISOと略す。）1603、1604と二つのインターリーブ1601、1602とインターリーブによるインターリーブング順序の逆の処理を行う二つのデインターリーブ1606、1607を具備して構成される。分離器1605はパリティ系列1、2をそれぞれ対応するSISO

0に分配するものであり、判定器1608は最終的に得られた軟出力データを二値に硬判定する為のものである。

【0006】図17は、RAMに蓄積されたインターリーブング順序（インターリーブパターンテーブルとして蓄積されている。）によりビット単位に並び替えを行う従来のインターリーブの例を示しており、インターリーブングを行うデータ系列1701は、インターリーブパターンテーブルによるインターリーブング順序が蓄積されたRAM1702によりデータ系列内のビット順序の入れ替えが行われ、インターリーブング後のデータ系列1703を得ている。

【0007】インターリーブング順序を出力するRAM1702の出力とインターリーブパターンテーブルとの関係は、同図1702に示すように、素数 p をベースとした長さ p で R 個のブロックとしたパターンテーブルで、矢印の様に縦方向の順に0, 8, 4, 12, 2, ...の順で読み出してインターリーブング後のデータ系列1703を得る様になっている。

【0008】

【発明が解決しようとする課題】IMT2000(W-CDMA)の標準規格3G TS25.212 V3.3.0(2000-06)に*

$$s(j)=[\nu \cdot s(j-1)] \bmod p, j=1, 2, \dots, (p-2). \text{ And } s(0)=1. \text{-----}(1)$$

これをテーブルにする。次に有限体の標数から1を引いた数 $p-1$ と互いに素なる数 $q(i)$ を R 個求める。最後に行を単位として行の順序入れ替えを行う(inter-row permutation処理)。行単位の順序入れ替えは、予め決められたパターン $T(i)$ に基づき行われる。これらのパ

$$U_i(j)=s([j \cdot r_i] \bmod (p-1)), j=0, 1, 2, \dots, (p-2), \text{ and } U_i(p-1)=0, \text{-----}(2)$$

ここで、 $U_i(j)$ は入れ替え前のビットポジションを示し、行入れ替え前 i 番目の行における行内順序入れ替え後 j 番目の出力位置に相当する。また、 $r_i(i)=q(i)$ であり、 $T(i)$ は上記で定義された入れ替え前の行位置で i 番目の行位置になる。

【0011】一例としてデータ長 $K=257$ の場合について説明する。二次元配列の行数 R を20としたときに素数 p で表せる列数 p を求めると、 $257/20=12.85$ であるから、それ以上で最も近い素数 p は、 $p=13$ となる。標数が13の有限体上の原始元 ν は2である。

【0012】そこでこの原始元 $\nu=2$ を用いて行内順序入れ替えを行う為のベースシーケンス $S(j)$ を(1)式から求めると、 $p=13$ 、 $\nu=2$ の場合
 $\{s(j)\}=\{1, 2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7, 0\}$
 となる。尚、最後に0を挿入する。

【0013】次に有限体の標数から1を引いた数 $p-1$ と互いに素なる数 $q(i)$ を $R(=20)$ 個求める。 $p=13$ の上述の例では、
 $\{q(i)\}=\{1, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79\}$

*よれば、各種のマルチメディアサービスに対応する為、インターリーブ長が1ビット毎の40ビットから5114ビット迄で、5075種類のインターリーブングパターンを用意する必要がある。この全てのインターリーブ長に対応するパターンテーブルを備える為には、膨大なメモリ量が必要となり現実的でない。そこで全ての種類のパターンを蓄積するのではなく、各インターリーブング長に応じて予め決められた演算に従いパターンを生成する方法が3G TS25.212 V3.3.0(2000-06) 4.2.3.2.3章に開示されている。

【0009】上記「3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and channel coding(FDD) (Release 1999) 3G TS25.212 V3.3.0(2000-06)またはV4.0.0(2000-12) 4.2.3.2.3章 Turbo code internal interleaver 16頁〜20頁」に開示されている素数インターリーブでは、データ長を、素数 p をベースとした長さ p で R 個のブロックとし、標数が p の有限体上の原始元 ν を用いて行内順序入れ替えを行う為のベースシーケンス $S(j)$ を以下の様にして求めている(intra-row permutation処理)。

※ターンとしては、自由距離を大きくする行間の交錯パターンが用いられている。

【0010】例えば、 i 番目の行内順序入れ替えを行う場合、次の様な処理を行う。

30★となる。

【0014】最後に(2)式により、予め決められたパターンに基づき行を単位として行の順序入れ替え(inter-row permutation処理)が行われるが、 $R=20$ の場合の自由距離を大きくする行間の交錯パターンは、
 $Pat1: \{T(i)\}=\{19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11\}$
 となる。

【0015】また、 $r_i(i)=q(i)$ であるから、

$$r_1(1)=q(1)=1=r_{19}$$

$$r_1(2)=q(2)=7=r_9$$

$$r_1(3)=q(3)=11=r_{14}$$

:

$$r_1(19)=q(19)=73=r_{15}$$

$$r_1(20)=q(20)=79=r_{11}$$

となるので、これらの値を(2)式に代入して $U_i(j)$ を求める。

【0016】上記従来技術では、この $U_i(j)$ をDSP(digital signal processor)等のソフトウェア処理によって計算し、図17に示す大規模RAM1702等へ転送してインターリーブ処理を行っている。

★50

【0017】一方ターボデコードの場合、反復復号を行うことになるが、例えば、8イタレーションの構成で2Mbpsの受信データ系列の復号を行う場合、上述のインターリービング順序へのアクセスは数10MHzという高速動作が要求される。これに対応するには、上述の演算に従って生成されたパターンを一度高速メモリに蓄え、そのメモリが数10MHzのアクセスを受け持つ構成にする必要がある。

【0018】しかしその為に必要なメモリ(RAM)の容量は、 5114×13 ビット=66482ビットを要し、ターボデコードを構成する要素の内、可成りの部分を占めるに至っている。更に、上述の演算に従って生成されたパターンを実際に処理を行っているターボデコード内のインターリービング用RAMに転送する必要があるが、このインターフェースには他のデータも同時に送る必要があり、インターフェース上のボトルネックを生じる様になった。

【0019】更に、可変レート機能を持たせた場合、頻繁にインターリーブ長の変更が生じるが、その場合更にインターフェース上のボトルネックを助長する様になり、マルチメディアサービスにおける転送レートに追随出来ないといった問題を生じるに至った。

【0020】このように、各種のマルチメディアサービスに対応した移動体通信システムに用いられるターボデコードの内部インターリーブは多様なインターリーブ長に対応出来る様にする必要があるため、各種のインターリービングパターンを用意する必要がある膨大なメモリ量を必要とする。更に高速データに対応する為にはインターリービングパターンを一度高速メモリに蓄える必要があり、その為の高速なメモリ容量を必要とし、これが回路規模の増大を招いていた。更に、可変レート機能を有したサービスにおいてはパラメータ転送によるインターフェースの輻輳を生じていた。

【0021】本発明の目的は、以上の問題点に鑑みなされたものであり、インターリービング順序発生器、インターリーブ、ターボエンコード並びにターボデコードにおいて、マルチメディアサービスにおける多様なインターリーブ長及びその転送レートに対し、少ないインターリーブ用RAM容量で実現し、しかもインターフェースにかかる負担が少なく、更に可変レート機能を持たせた場合でもマルチメディアサービスにあった転送レートに追随出来る手段を提供することにある。

【0022】

【課題を解決するための手段】請求項1に記載された本発明は、データ長を、素数 p をベースにした長さ p で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成する手段を有し、標数が p の有限体の元を原始元 α に対し前記 $q_0, q_1, q_2, \dots, q_{R-1}$ の冪乗としてそれぞれ $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ のごとく生成及び記憶する手段を有

し、第0の順序入れ替えにあたっては、前記ブロック入れ替えを行う為の予め決められたパターンを生成或いは記録する手段からの出力を p 倍した値に1を順次足し合わせて行い、第 j の順序入れ替えにあたっては前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を有限体上で j 乗してそれぞれ $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ のごとく生成する手段を用いて上記同様に p 倍した値に前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を順次足し合わせを $j=1 \sim (p-2)$ において繰り返すインターリービング順序発生器をその特徴としている。

【0023】請求項2に記載された本発明は、請求項1に記載されたインターリービング順序発生器において、前記 $j=1 \sim (p-2)$ の繰り返しに当たっては前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を生成及び記憶する手段と、この記憶した値を順次有限体における高速乗算器に入力することによって前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ の値を逐次的に更新することを特徴としている。

【0024】請求項3に記載された本発明は、請求項1または2に記載のインターリービング順序発生器において、前記 $j=p-1$ において $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ に相当する値を全て0とすることを特徴としている。

【0025】請求項4に記載された本発明は、データ長を、素数 p をベースにした長さ $p-1$ で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成する手段を有し、標数が p の有限体の元を原始元 α に対し前記 $q_0, q_1, q_2, \dots, q_{R-1}$ の冪乗としてそれぞれ $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ のごとく生成及び記憶する手段を有し、第0の順序入れ替えにあたっては、前記ブロック入れ替えを行う為の予め決められたパターンを生成或いは記録する手段からの出力を $p-1$ 倍した値に1を順次足し合わせて行い、第 i の順序入れ替えにあたっては前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を有限体上で j 乗してそれぞれ $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ のごとく生成する手段を用いて上記同様に $p-1$ 倍した値に前記 $(\alpha^{q_0})^j, (\alpha^{q_1})^j, (\alpha^{q_2})^j, \dots, (\alpha^{q_{R-1}})^j \pmod{p}$ を順次足し合わせを $j=1 \sim (p-2)$ において繰り返すインターリービング順序発生器をその特徴としている。

【0026】請求項5に記載された本発明は、請求項4に記載のインターリービング順序発生器において、前記順次足し合わせを行った値に対し1を減算することを特徴としている。

【0027】請求項6に記載された本発明は、請求項4または5に記載のインターリービング順序発生器において、前記 $j=1 \sim (p-2)$ の繰り返しに当たっては前記 $\alpha^{q_0}, \alpha^{q_1}, \alpha^{q_2}, \dots, \alpha^{q_{R-1}} \pmod{p}$ を生成及び

記憶する手段と、この記憶した値を順次有限体における高速乗算器に入力することによって前記 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ の値を逐次的に更新することを特徴としている。

【0028】請求項7に記載された本発明は、データ長を、素数 p をベースにした長さ $p+1$ で R 個のブロックとし、 $p-1$ とは互いに素なる R 個の異なる整数 $q_0, q_1, q_2, \dots, q_{R-1}$ を生成する手段を有し、標数が p の有限体の元を原始元 ℓ に対し前記 $q_0, q_1, q_2, \dots, q_{R-1}$ の冪乗としてそれぞれ $\ell^{\wedge}q_0, \ell^{\wedge}q_1, \ell^{\wedge}q_2, \dots, \ell^{\wedge}q_{R-1} \pmod{p}$ のごとく生成及び記憶する手段を有し、第0の順序入れ替えにあたっては、前記ブロック入れ替えを行う為の予め決められたパターンを生成或いは記録する手段からの出力を $p+1$ 倍した値に1を順次足し合わせて行い、第 j の順序入れ替えにあたっては、前記 $\ell^{\wedge}q_0, \ell^{\wedge}q_1, \ell^{\wedge}q_2, \dots, \ell^{\wedge}q_{R-1} \pmod{p}$ を有限体上で j 乗してそれぞれ $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ のごとく生成する手段を用いて上記同様に $p+1$ 倍した値に前記 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ を順次足し合わせを $j=1 \sim (p-2)$ において繰り返すインターリービング順序発生器をその特徴としている。

【0029】請求項8に記載された本発明は、請求項7に記載のインターリービング順序発生器において、前記 $j=1 \sim (p-2)$ の繰り返しに当たっては前記 $\ell^{\wedge}q_0, \ell^{\wedge}q_1, \ell^{\wedge}q_2, \dots, \ell^{\wedge}q_{R-1} \pmod{p}$ を生成及び記憶する手段と、この記憶した値を順次有限体における高速乗算器に入力することによって前記 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ の値を逐次的に更新することを特徴としている。

【0030】請求項9に記載された本発明は、請求項7または8に記載のインターリービング順序発生器において、前記 $j=p-1$ において $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ に相当する値を全て0とすることを特徴としている。

【0031】請求項10に記載された本発明は、請求項7～9のいずれかに記載のインターリービング順序発生器において、前記 $j=p$ において $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ に相当する値を全て p とすることを特徴としている。

【0032】請求項11に記載された本発明は、請求項1～10のいずれかに記載のインターリービング順序発生器において、前記インターリービング順序発生器からの出力信号がインターリーバ対象範囲を超えた場合、該信号をスキップして次の該範囲内の信号を使用することを特徴としている。

【0033】請求項12に記載された本発明は、請求項1～11のいずれかに記載のインターリービング順序発生器において、前記有限体の高速乗算器を複数個有し、前記 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j$

$j \pmod{p}$ の値の更新を前記 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots, (\ell^{\wedge}q_{R-1})^{\wedge}j \pmod{p}$ の内複数個同時に実行し、前記インターリービング順序発生器からの出力がインターリーバ対象範囲を超えた場合、該信号をスキップして次の範囲内の信号を使用し、淀みなく信号を発生するようにしたことを特徴としている。

【0034】請求項13に記載された本発明は、請求項12に記載のインターリービング順序発生器において、前記有限体における高速乗算器を2個備え、前記生成及び記憶した $\ell^{\wedge}q_0, \ell^{\wedge}q_1, \ell^{\wedge}q_2, \dots, \ell^{\wedge}q_{R-1} \pmod{p}$ を、偶数乗数と奇数乗数に分割した $\ell^{\wedge}q_0, \ell^{\wedge}q_2, \ell^{\wedge}q_4, \dots \pmod{p}$ と $\ell^{\wedge}q_1, \ell^{\wedge}q_3, \ell^{\wedge}q_5, \dots \pmod{p}$ に対して、前記2個の高速乗算器を割り当て、前記有限体上で j 乗することにより、 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots \pmod{p}$ と $(\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_3)^{\wedge}j, \dots \pmod{p}$ の値を、並行して同時に更新することを特徴としている。

【0035】請求項14に記載された本発明は、請求項1～13のいずれかに記載のインターリービング順序発生器出力をデータが蓄積されたメモリのアドレス信号としてデータの読み出しを行うことにより順序入れ替えを行うインターリーバをその特徴としている。

【0036】請求項15に記載された本発明は、請求項1～13のいずれかに記載のインターリービング順序発生器出力をデータが蓄積するメモリのアドレス信号としてデータを書き込むことにより順序入れ替えを行うインターリーバをその特徴としている。

【0037】請求項16に記載された本発明は、請求項14に記載のインターリーバを、ターボエンコードの内部インターリーバとするターボエンコードをその特徴としている。

【0038】請求項17に記載された本発明は、請求項14または15に記載のインターリーバの内少なくとも一方をターボデコードの内部インターリーバとし、他方を内部デインターリーバとするターボデコードをその特徴としている。

【0039】請求項18に記載された本発明は、請求項1～13のいずれかに記載のインターリービング順序発生器出力を、データが蓄積されたデュアルポートメモリの読み出し用アドレス信号として、前記データ内容の読み出しを行い、予め決められた値で遅延を施した該アドレス信号を書き込み用アドレス信号としてデータ内容の書き込みを行うことによりターボデコードの内部インターリーバと内部デインターリーバを同時に実現したターボデコードをその特徴としている。

【0040】本発明によれば、各種のマルチメディアサービスに対応した移動体通信システムにおいて、ターボデコードに用いられている素数体をを用いたインターリーバで多様なインターリーブ長に回路規模の増大を招くことなく対応することができ、また、インターリービング順序発生器、インターリーバ、ターボエンコード並びに

ターボデコードにおいて、少ないインターリーブ用RAM容量で実現することができ、更に、インターフェースにかかる負担が少なくなるので、可変レート機能を有したマルチメディアサービスであっても転送レートに容易に追従することが出来るようになる。

【0041】

【発明の実施の形態】次に、本発明の実施形態について図面を参照しながら説明する。

【0042】図13は、本発明のインターリーブ順序発生器を用いたターボ符号器（ターボエンコーダ）の実施形態を示すブロック構成図である。図13に示すターボエンコーダの動作については後述するが、このターボエンコーダと図15に示す従来のターボエンコーダとの主な差は、インターリーブ1501にある。

【0043】即ち、従来のインターリーブは図17に示す様に、インターリーブ順序を蓄積した大規模なRAM1702を必要としていた。これに対し図13に示す実施形態においては、インターリーブ順序発生器1301を用いることにより、インターリーブ順序を蓄積した大規模なRAMを使うことなく素数インターリーブを実現したことを特徴としている。

【0044】本発明で採用しているインターリーブ方法も、基本的には上記文献に記載の素数インターリーブ*

$$U_i(j) = \{(\nu^{\{j \cdot r_i - n \cdot (p-1)\}}) \bmod p = (\nu^{r_i})^j \cdot (\nu^{(p-1)})^{(-n)} \bmod p \\ = \{(\nu^{r_i})^j \bmod p\} \cdot \{(\nu^{(p-1)})^{(-n)} \bmod p\} \bmod p$$

ここでFermat's Theoremより全ての要素aに対して

$$a^{(p-1)} \equiv 1 \pmod{p}, \text{ where } p: \text{prime}$$

が成り立つから、 $(\nu^{(p-1)})^{(-n)} \bmod p = 1$

従って上式は、

$$U_i(j) = (\nu^{r_i})^j \bmod p \text{-----}(3)$$

となる。

【0047】ここで上述より $r_i(i) = q_i$ であり、 $R = 20$ の場合における前記行間交錯パターン

Pat1: {T(0), T(1), ..., T(R-1)} = {19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15, 11}

を例にとって説明すると、0行目 ($i = 0$) にくる行位置は $T(0) = 19$ 行目で $q_0 (= r_{19})$ がその行の値として選ばれる。同様に1行目 ($i = 1$) には $T(1) = 9$ 行目が来て $q_1 (= r_9)$ がその行の値として選ばれる。

【0048】この様に各行の r_i が異なる値に設定され、その結果、(3)式の (ν^{r_i}) が各行で異なる値となり、各行における行内順序入れ替えが行毎に異なり、ランダム化される事になるのである。また、 $r_i(i) = q_i$ で与えられる q_i は上述より $(p-1)$ と互いに素なる関係で選ばれている。 ν は原始元であるから p を法としてその位数は $p-1$ である。

【0049】この p を法として構成される集合の任意の要素を a とすると $p-1 \bmod p$ での最大の位数であるから、

$$a^{(n \cdot (p-1))} \equiv 1 \pmod{p}$$

*バと同様であるが、本発明においては、従来の様に予め $U_i(j)$ を計算してRAM等に転送しておくのではなく、実時間で $U_i(j)$ を発生してインターリーブ処理を行う構成としたことをその特徴としており、そのため、従来必要としていた大規模RAMを不要とすることができる。以下その方法を説明する。

【0045】先ず、上記(1)式より、

$$s(0) = 1$$

$$s(1) = \nu \bmod p$$

$$s(2) = \nu^2 \bmod p$$

:

$$s(j) = \nu^j \bmod p$$

となる。なお、 $\nu^j \equiv \nu^j$ である。上式における ν は原始元であるから、 $\bmod p$ 上で繰り返し乗算処理することにより、 $\bmod p$ 上で構成される有限体の全ての要素を網羅することになる。

【0046】この結果を(2)式に適用すると、

$$U_i(j) \equiv s(\{j \cdot r_i\} \bmod (p-1)) = \{(\nu^{\{j \cdot r_i\} \bmod (p-1)})\} \bmod p$$

ここで、

$$\{j \cdot r_i\} \bmod (p-1) = j \cdot r_i - n \cdot (p-1)$$

と置き換えると、

※なる関係が成り立つ。

【0050】従って、(3)式を

$$(\nu^{r_i})^j \equiv 1 \pmod{p}$$

とする為の条件は

$$r_i \cdot j \equiv n \cdot (p-1) \pmod{p-1}$$

となり、 $(p-1) \mid r_i \cdot j$ となるが必要となる。

【0051】ところが、 r_i と $(p-1)$ は互いに素であるから、 r_i の中に $(p-1)$ を構成する因数は存在せず、結局 $(p-1) \mid j$ となること即ち、 (ν^{r_i}) の位数が $(p-1)$ であることを示しており (ν^{r_i}) も p を法として原始元であることに他ならない。

【0052】従って、

$$U_i(j) = (\nu^{r_i})^j \bmod p \quad \text{where } j = 0, 1, 2, \dots, (p-2).$$

40 は各行で異なる原始元 (ν^{r_i}) を乗数とする線形合同法の一種である乗算合同法によるランダム系列発生アルゴリズムを構成していることに他ならない。 (ν^{r_i}) は p を法とする素体の原始元であるから、その冪乗の形で表せる $(\nu^{r_i})^j$ は素体の全ての要素を網羅し、インターリーブに必要な一対一の写像関係を維持出来る様になっている。

【0053】即ちこのことは、(3)式における (ν^{r_i}) を乗数として逐次乗算することにより(1)式

$$s(j) = [\nu \cdot s(j-1)] \bmod p, \quad j = 1, 2, \dots, (p-2), \text{ and } s(0) = 1.$$

※50

によって生成されるテーブルを持たなくとも $U_i(j)$ が得られることを示している。

【0054】例えば、データ長 $k=5114$ ビットの場合でみると、二次元配列の行数20に対して素数で表せる列数 p は、 $5114/20=255.7$ で最も近い素数 $p=257$ が列数となる。これだけで比較したとしても、

$s(j)=[r \cdot s(j-1)] \bmod p$, $j=1, 2, \dots, (p-2)$, and $s(0)=1$.

によって生成されるテーブルは257個必要になるのに対して、本発明では

$(r^i \bmod p)$, $i=0, \dots, 19$

の20個で済むことになる。即ち、メモリを10分の1以下に削減することが可能になる。

【0055】上記より分かる様に、本発明は行数が同じであればデータ長が長いほど効果がある。一方、ターボ符号にはインターリーブ利得(interleaver gain)と呼ばれる特徴があり、データ長が長ければ長い程高い符号化利得を得られる。即ち本発明はターボ符号に適した方法といえる。尚以下の実施例では解説のし易さから短いデータ長を例にとって説明しているが、データ長は任意の長さを採り得る。

【0056】図1は、本発明のインターリーブ順序発生器において、上記(3)式の $U_i(j)$ を生成するブロックである、 $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ 生成部の第1の実施形態を示すブロック図である。本実施例ではデータ長 $K=257$ の場合を示しており、二次元配列は素数 $p=13$ 、 $R=20$ によって表わされる。

【0057】上述の説明で用いている (r^{r_i}) 入れ替え後の0行目を見ると $(r^{r_{19}})=(r^{q_0})$ となる。同様に入れ替え後の1行目を見ると、 $(r^{r_9})=(r^{q_1})$ となる。即ち入れ替え後の $i=0 \sim 19$ 行目に対して $(r^{r_{T(i)}})=(r^{q_i})$ となる。これは、入れ替え後の行番号 i に対しての乗数を (r^{q_i}) とすればよいことを示している。

【0058】但しここで注意を要するのは、この関係は行内部で成り立つ関係であって、行入れ替え前の行位置を加算する必要がある事を忘れてはならない。この場合、 $T(i)$ 行から i 行へ入れ替えられたのであるから、列の数 $p=13$ とすると、 $p \times T(i)$ を加算する必要がある。

【0059】図1を見ると乗数 $r^{q_0}, r^{q_1}, r^{q_2}, \dots, r^{q_{R-1}} \bmod p$ を格納するレジスタ101にセクタ104を介して有限体上の高速乗算器103が接続されている。この乗算器出力はセクタ105を介して乗算結果を一次保存するレジスタ102に接続されている。レジスタ102の出力はセクタ106を介して出力されるとともに前記乗算器103のもう一つの入力へ接続されている。

【0060】セクタ104、105、及び106はそれぞれ連動して選択するように制御されており、それぞ

れ行番号 $i=0 \sim R-1$ に対応してセクタ104は $(r^{q_0})^i \bmod p \sim (r^{q_{R-1}})^i \bmod p$ を、セクタ105と106は $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ を選択する様になっている。

【0061】尚上記セクタをアドレス制御に置き換え、上記レジスタをRAMに置き換えて同様の構成を実現することも可能である。以下、図1を参照しながら本発明のインターリーブ順序発生動作について説明する。

【0062】先ず第0の順序入れ替え時の動作を説明する。レジスタ102の初期値は全て‘1’にプリセットされる構成になっている。 $j=0$ に相当する第0の順序入れ替えから $j=1$ に相当する第1の順序入れ替えの遷移時は、セクタ106が選択した値は全て‘1’である。この値が乗算器103の入力の一方に入ると同時に出力端107から出力される。即ち $j=0$ に相当する第0順序入れ替えに当たっては出力端107の値は全て‘1’となる。

【0063】この時、セクタ104は $(r^{q_0}) \bmod p \sim (r^{q_{R-1}}) \bmod p$ を順次選択していく。従って乗算器103の出力は $(r^{q_0})^1 \bmod p \sim (r^{q_{R-1}})^1 \bmod p$ となり、連動して動作するセクタ105によってレジスタ102には、 $(r^{q_0})^1 \bmod p \sim (r^{q_{R-1}})^1 \bmod p$ が初期値‘1’に代わって順次更新されることになる。

【0064】次に $j=1$ に相当する第1の入れ替えにあたっては、セクタ106が選択する値は $(r^{q_0})^2 \bmod p \sim (r^{q_{R-1}})^2 \bmod p$ である。この値が乗算器103の入力の一方に入ると同時にこの $(r^{q_0})^1 \bmod p \sim (r^{q_{R-1}})^1 \bmod p$ が出力端107から送出される。

【0065】この時セクタ104は $(r^{q_0})^2 \bmod p \sim (r^{q_{R-1}})^2 \bmod p$ を順次選択していくから、乗算器103の出力は $(r^{q_0})^2 \bmod p \sim (r^{q_{R-1}})^2 \bmod p$ となり、連動して動作するセクタ105によってレジスタ102には、 $(r^{q_0})^2 \bmod p \sim (r^{q_{R-1}})^2 \bmod p$ が入力され、 $(r^{q_0})^1 \bmod p \sim (r^{q_{R-1}})^1 \bmod p$ に代わって順次更新されることになる。

【0066】同様に動作を進めていくと、第 j の入れ替えにあたっては、セクタ106が選択する値は $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ である。この値が乗算器103の入力の一方に入ると同時にこの $(r^{q_0})^{j-1} \bmod p \sim (r^{q_{R-1}})^{j-1} \bmod p$ が出力端107から送出される。この時セクタ104は $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ を順次選択していくから、乗算器103の出力は $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ となり、連動して動作するセクタ105によってレジスタ102には、 $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ が入力され、 $(r^{q_0})^{j-1} \bmod p \sim (r^{q_{R-1}})^{j-1} \bmod p$ に代わって順次更新されることになる。

【0067】この様にして生成された $(r^{q_0})^j \bmod p \sim (r^{q_{R-1}})^j \bmod p$ は(3)式の $U_i(j)=(r^{r_i})^j \bmod p$

mod p において、行間入れ替えを行った後の二次元配列の列方向に向かって読み出した値に他ならない。

【0068】従って、既に説明したように行入れ替え前の行位置を加算する必要がある。即ち、 $T(i)$ 行から i 行へ入れ替えられたとすると、 $p \times T(i)$ を加算する必要がある。

【0069】図2は、本発明のインターリービング順序発生器の第1実施形態を示すブロック図である。

【0070】同図において、201は、上記 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部である。また205は、ブロック入れ替えパターン $T(i)$ が予め記憶されたテーブルであり、行の更新に合わせて $T(i)$ 、 $(i=0 \sim R-1)$ を出力する。

【0071】セクタ204は $j=0 \sim p-2$ の間は $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部201を選択しているが、最後の $j=p-1$ になった時、零を出力する零出力部202を選択するように動作する。従って、最後の列については $p \times T(i)$ における $i=0 \sim R-1$ の値がインターリービング順序出力として出力端209より送出される。

【0072】二次元配列の列数を設定する列数設定部206の設定値は本実施例の場合 p となっており、乗算器207によって $p \times T(i)$ が生成される。この値と上述のセクタ204によって選ばれた値が加算器208によって加えられる。

【0073】 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部201の $i=0$ 時の値 $(\nu^{q_0})^j \pmod{p}$ から $i=R-1$ 時の値 $(\nu^{q_{R-1}})^j \pmod{p}$ への各遷移タイミングは、テーブル205がブロック入れ替えパターン $T(i)$ を $i=0$ から $i=R-1$ まで出力する各遷移タイミングと同期しており、その結果加算器208の出力は行間入れ替えを行った後の $U_i(j) = (\nu^{r_i})^j \pmod{p}$ による二次元配列を列方向に向かって読み出した値となる。

【0074】次に、データ長 $K=280$ の場合を説明する。二次元配列の行数 R を20とする。 $280/20=14$ であるからそれ以上で最も近い素数を選ぶところであるが、列数 $C=p+1$ として素数 $p=13$ でも二次元配列を構成することが出来る。そこで、 14×20 の二次元配列への適用を考える。 $C=p+1$ の場合も原始元を使用することに変わりはない。票数が13の有限体上の原始元は2である。この原始元 $\nu=2$ を用いて行内順序入れ替えを行う為の式を以下に示す。

【0075】行内順序入れ替えを行う為の式は、 $U_i(j) = (\nu^{r_i})^j \pmod{p}$ where, $j=0, 1, 2, \dots, (p-2)$,

となり、上式も既に説明した列数を p とした場合と同様の理由から導き出せる。以下、 $C=p+1$ の場合の本発明のインターリービング順序発生器について説明する。

【0076】前述の $C=p$ の場合と同様に入れ替え後の $i=0 \sim 19$ 行目に対して $(\nu^{r_{T(i)}})^j = (\nu^{q_i})^j$ となるか

ら、入れ替え後の行番号 i に対しての乗数を $(\nu^{q_i})^j$ とすればよい。 $T(i)$ 行から i 行へ入れ替えられたのであるから、列の数 $p+1=13+1=14$ として、 $(p+1) \times T(i)$ を加算することが必要なことも同様である。

【0077】この処理は図2のインターリービング順序発生器で行われる。同図において、図1で生成される $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ は、 $C=p+1$ の場合も $C=p$ の場合と同様に $i=0 \sim p-2$ において同じ処理で実現出来ることになる。セクタ204は $j=0 \sim p-2$ の間 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部201を選択しているが、 $j=p-1$ になった時、零出力部202を選択する様に動作する。従ってこの時はテーブル205のブロック入れ替えパターン $T(i)$ と列数設定部206の設定値 $C=p+1$ とを乗算器207により乗算した結果が加算器208を通して出力端209に出力されることになる。

【0078】最後の $j=p$ になった時、セクタ204は p 設定部203の設定値 p を選択するように動作する。従って、最後の列については $(p+1) \times T(i)$ における $i=0 \sim R-1$ の値と p の和がインターリービング順序出力として209より送出される。ここで、テーブル205は、行の更新に合わせてブロック入れ替えパターン $T(i)$ 、 $i=0 \sim R-1$ を出力する。

【0079】この場合、列数設定部206は二次元配列の列数設定値を $p+1$ に設定しており、乗算器207によって $(p+1) \times T(i)$ が生成され、この値と上述のセクタ204によって選択された値が加算器208によって加えられる。

【0080】 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部201の $i=0$ 時の値 $(\nu^{q_0})^j \pmod{p}$ から $i=R-1$ 時の値 $(\nu^{q_{R-1}})^j \pmod{p}$ への各遷移タイミングは、205の $T(i)$ で $i=0$ から $i=R-1$ を出力する遷移タイミングと同期しており、その結果加算器208の出力は行間入れ替えを行った後の $U_i(j) = (\nu^{r_i})^j \pmod{p}$ による二次元配列を列方向に向かって読み出した値となる関係は $C=p$ の場合と同様である。

【0081】次に、データ長 $K=320$ の場合の例を説明する。二次元配列の行数 R を20とする。 $320/20=16$ であるから、それ以上で最も近い素数 $p=17$ となる。しかし列数 $C=p-1=16$ 行としても二次元配列を構成することが出来る。そこで、 16×20 の二次元配列への適用を考える。 $p-1$ の場合も原始元を使用することに代わりはない。票数が17の有限体上の原始元は3である。

【0082】この原始元 $\nu=3$ を用いて行内順序入れ替えを行う為の式を以下に示す。

$U_i(j) = (\nu^{r_i})^j \pmod{p}$ where, $j=0, 1, 2, \dots, (p-2)$,

上式は既に説明した列数を p とした場合と同様の理由から導き出せる。

【0083】以下、図1を参照しながら本発明のインターリーピング順序発生器を $C=p-1$ に適用した場合について説明する。前述の $C=p$ の場合と同様に入れ替え後の $i=0\sim 19$ 行目に対して $(\nu^{qr(i)})=(\nu^{qi})$ となるから、入れ替え後の行番号 i に対しての乗数を (ν^{qi}) とすればよい。 $T(i)$ 行から i 行へ入れ替えられたのであるから、列の数 $p-1=17-1=16$ として、 $(p-1)\times T(i)$ を加算することが必要なことも同様である。

【0084】この処理は図2のインターリーピング順序発生器で行われている。従って、図1で生成される $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ は $C=p$ の場合と同様に $i=0\sim p-2$ において $C=p-1$ の場合も同じ処理で実現出来ることになる。ただし $C=p$ の場合には、 $i=p-1$ になった時零出力部202を選択する様に動作するが、 $C=p-1$ の場合には $i=0\sim p-2$ 迄であるので、セクタ204は $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部201を選択したままであり、零出力部202や p 出力部203を選択することはない。また $C=p-1$ の場合、零出力部202の選択が無い為、順序入れ替えパターンが $1\sim C$ となる。そこで、入れ替えパターンを $0\sim C-1$ に合わせる為に発生した値に対し1を減算する構成も可能である。

【0085】図3は、 $C=p-1$ の場合におけるインターリーピング順序発生器の構成例を示すブロック図である。 $C=p-1$ の場合であるから、セクタ204に相当する動作は必要ない。

【0086】 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部301が直接加算器308に入力される。この加算器308には減算の為に-1の値を出力する定数発生部310から-1が入力されている。また、列数設定部306からの二次元配列の列数 $p-1$ とテーブル305のブロック入れ替えパターン $T(i)$ が乗算器307で掛け算され、その結果の $(p-1)\times T(i)$ が加算器308に入力される。これらの加算結果がインターリーピング順序出力として出力端309より送出される。

【0087】図4は、本発明のインターリーピング順序発生器において、上記(3)式の $U_i(j)$ を生成するブロックである。 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部の第2の実施形態を示すブロック図である。

【0088】インターリーピング順序発生器からの出力信号がインターリーブ対象範囲を超えた場合、その信号をスキップすることになるが、本実施形態では、そのようなスキップが生じた場合であっても淀みなく信号を発生させるために、上述した有限体の高速乗算器を二つ用意している。

【0089】図4において、図1における乗数 ν^{q_0}, ν

$\nu^{q_1}, \nu^{q_2}, \dots, \nu^{q_{R-1}} \pmod{p}$ を格納するレジスタ101が二分割され401と411となっているがトータルの容量は図1の場合と変わらない。分割方法としては色々な方法が可能であるが、ここでは乗数 $\nu^{q_0}, \nu^{q_2}, \nu^{q_4}, \dots \pmod{p}$ を格納する偶数用レジスタ401と乗数 $\nu^{q_1}, \nu^{q_3}, \nu^{q_5}, \dots \pmod{p}$ を格納する奇数用レジスタ411に分割した例について説明する。

【0090】有限体上の高速乗算器は403と413の二つであり、それぞれセクタ404及び414を通じて乗数が格納されているレジスタから一方の乗算器入力を得る様になっている。この乗算器出力はそれぞれセクタ405及び415を介して乗算結果を一時保存するレジスタ402及び412に接続されている。

【0091】レジスタ402、412の出力は、それぞれセクタ406、416を介して出力端407および417へ出力されるとともに前記乗算器403及び413のもう一つの入力へ接続されている。セクタ404、405及び406とセクタ414、415及び416はそれぞれ連動して選択動作を行う様に制御されており、その結果図1で説明したものと同一計算結果を偶数と奇数に対して同時に得ることが出来る。即ち $(\nu^{q_0})^j, (\nu^{q_2})^j, \dots \pmod{p}$ と $(\nu^{q_1})^j, (\nu^{q_3})^j, \dots \pmod{p}$ が同時に得られる。

【0092】尚、セクタをアドレス制御に置き換え、更にデュアルポートRAMにより偶数に相当するアクセスと奇数に相当するアクセスを同時に実行する様に構成し、一つのRAMで二つのレジスタと同じ動作を実現する様に構成することも可能である。

【0093】図5～図6は、本発明のインターリーピング順序発生器の第2実施形態を示すブロック図である。

【0094】本実施形態は、図4に示す二つに分割された $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部を備えることにより、インターリーブ対象範囲を超える信号のスキップを行うように構成されており、図5に示すブロックで図2に相当する列数 \times ブロック入れ替えパターン $T(i)$ の加算処理が行われ、図6に示すブロックでインターリーブ対象範囲を超える信号のスキップ動作が実行される。

【0095】図5において、 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部は、 $(\nu^{q_0})^j, (\nu^{q_2})^j, \dots \pmod{p}$ 生成部501と、 $(\nu^{q_1})^j, (\nu^{q_3})^j, \dots \pmod{p}$ 生成部511とに分割されているが、これらの基本的な動作は図2の、 $(\nu^{q_0})^j \pmod{p} \sim (\nu^{q_{R-1}})^j \pmod{p}$ 生成部の動作と同様であるのでその詳細動作説明は省略する。

【0096】列数設定部506とブロック入れ替えパターン $T(i)$ 発生部505は偶数と奇数で共用できるので一つで構成し、加算器508によって偶数に対応するインターリーピング順序を偶数出力端509から出力し、加算器518によって奇数に対応するインターリー

ピング順序を奇数出力端519から出力している。

【0097】これらのインターリーピング順序を表す信号に対しインターリーブ対象範囲を超える信号をスキップするのが図6のブロックである。同図において偶数に対応するインターリーピング順序信号が602から、奇数に対応するインターリーピング順序信号が601から入力する。これらの信号はそれぞれコンパレータ604と605でトータルビット数607と比較され、このトータルビット数以上のものがインターリーブ範囲外としてスキップされる。

【0098】この様にして生成されたインターリーブ範囲内のインターリーピング順序信号は切り替えスイッチ608によって元の順番に並び替えられFIFO609に入力される。FIFOの内容がいっぱいになるとFIFO609からはバッファフル(BUFFER FULL)信号が出力され、特に図示していないが、このバッファフル信号が各ブロックに対するホルト(HALT)信号610となって各ブロックの動作が一時停止する。

【0099】最終的なインターリーピング順序出力がFIFO609より読み出され、信号が端子611から出力されると、ホルト(HALT)信号が解除され各ブロックの動作が再開される。即ち、FIFO609のバッファリング機能によってインターリーピング順序出力611は淀みなく信号を発生することが出来る。

【0100】図7は、図1の有限体上の乗算器103や、図4の有限体上の乗算器403及び413の構成例を示すブロック図である。本実施例の有限体上の乗算器は、乗算701とモジュロ演算702とからなる二つの部分から構成される。モジュロ演算702は、図8に示す比較減算回路801によって構成され、図9に示すような演算901を実行する。

【0101】図9の演算901は、乗算器701の演算結果がバイナリーで1010010110000011であった時、 $p=10010011$ でモジュロを取る例を示している。先ず上位8ビットで比較減算されるが、これは比較減算回路801の構成で実現出来る。上位ビットMSBで比較結果を判断して p 以上の値ならば減算した値を出力する。同様の処理を1ビットずつシフトしながら最終的にモジュロ演算結果を得ることが出来る。

【0102】以上、本実施形態のインターリーピング順序発生器について説明したが、次に、このインターリーピング順序発生器を用いて実際にデータの順序入れ替えを行う処理について説明する。

【0103】図10は、RAMに蓄積されたデータの順序入れ替えを行うことによりインターリーピング処理を行う方法である。インターリーピング順序発生器1001からの信号をRAM1002のアドレス信号としてデータを読み出すことによりインターリーピングを行う。例えばインターリーピング順序発生器から0, 8, 4, 50

12, 2, . . . , 7, 15という系列がRAM1002の読み出しアドレス(RD Adr)に入力したとすると、アドレス順に並べられて蓄積されているデータの0番目のデータ、8番目のデータ、. . . がRAM1002から出力され順序入れ替えが行われる。

【0104】図11は、書き込みによりデインターリーピングを行う方法である。図10の読み出しによるインターリーピングと同様にインターリーピング順序発生器1101からの信号はRAM1102のアドレスとして入力している。図10と異なる点はこのアドレス信号は書き込み用アドレスであり、デインターリーピング後のデータはこのRAM1102に蓄積されることになる。

【0105】例えば上述のインターリーブされたデータが0番目のデータ、8番目のデータ、. . . の順でRAM1102に入力したとする。インターリーピング順序発生器1101からは図10と同じ0, 8, 4, 12, 2, . . . , 7, 15という系列がRAM1102の書き込みアドレス(WR Adr)に入力したとする。この結果、RAM1102には当初の順列に復元して、アドレス0, 1, 2, . . . に対し、0番目のデータ、1番目のデータ、2番目のデータ、. . . の順に復元され、デインターリーブが実行される。

【0106】尚、デインターリーブもインターリーブもパターンによって入れ替え可能であり、一方がインターリーブと呼ぶならばもう一方がデインターリーブに、逆に一方をデインターリーブと呼ぶならばもう一方はインターリーブとなる。この様に同じインターリーピング順序発生器を用いてインターリーブもデインターリーブも実現出来る。

【0107】図12は、この関係を用いてインターリーブとデインターリーブを同時に一つのインターリーピング順序発生器1201で実現したものであり、後述するターボデコードの外部情報系列と事前情報系列(アプリオリ)の入れ替え時に用いられるものである。

【0108】同図において、デュアルポートRAM1202にはアドレス順に受信シンボルに対応したアプリオリデータが蓄積されている。インターリーピングが行われた更新期間になるとインターリーピング順序発生器1201からインターリーピングパターンに応じて0, 8, 4, 12, 2, . . . , 7, 15といった系列がRAM1202の読み出しアドレスに入力される。これに応じてRAM1202からは0番目のデータ、8番目のデータ、. . . が出力される。

【0109】このインターリーブされたデータは、後述するターボデコードで処理されたあと元のデータ順序に戻すデインターリーブが必要になる。そこで、遅延器1203が挿入されターボデコードの処理時間分遅らせてデインターリーブ処理が行われる構成になっている。

【0110】例えば、処理されたデータが0番目のデータ、8番目のデータ、. . . の順でRAM1202に入

力したとする。インターリーピング順序発生器1201から遅延器1203を介した書き込み用アドレス信号は0, 8, 4, 12, 2, ..., 7, 15となってRAM1202にはアドレス0, 1, 2, ...に対し、0番目のデータ、1番目のデータ、2番目のデータ、...の順に復元されデインターリーブが実行される。

【0111】図13は、以上説明したインターリーピング順序発生器1301とそれを用いてインターリーブ処理を行う為のデュアルポートRAM1303を具備した本発明のターボエンコーダの実施形態を示すブロック図である。

【0112】本実施形態のターボエンコーダも、図15に示す従来のターボエンコーダと同様に、2つのコンポーネントエンコーダ1304と1305を有しており、コンポーネントエンコーダ1304にはインターリーブ処理を行わない情報系列が入力され、コンポーネントエンコーダ1305にはインターリーブ処理された情報系列が入力される。

【0113】そこで、デュアルポートRAM1303の一方のアドレス入力RD ADR1としてアップカウンタ1302の出力を入力し、もう一方のアドレス入力RD ADR2としてインターリーピング順序発生器1301の出力を入力する。そして、アップカウンタ1302からのアドレス入力RD ADR1により読み出されたデュアルポートRAM1303の情報系列をコンポーネントエンコーダ1304へ入力し、インターリーピング順序発生器1301からのアドレス入力RD ADR2により読み出されたデュアルポートRAM1303のインターリーブされた情報系列をコンポーネントエンコーダ1305へ入力する。

【0114】図14は、上述のインターリーピング順序発生器1402とそれを用いてインターリーブ処理及びデインターリーブ処理を行う為のデュアルポートRAM1407及び1406を具備した本発明のターボデコーダの実施形態を示すブロック図である。

【0115】情報系列が蓄積されているデュアルポートRAM1406の読み出し用アドレスにはアップカウンタ1401またはインターリーピング順序発生器1402が選択スイッチ1403を介して接続されている。ターボデコーディングの各イタレーション処理においてインターリーブを行わない処理に対応したパリティビット1によるデコードとインターリーブを施した処理に対応したパリティビット2によるデコードが存在する。選択スイッチ1403と1404はそれを切り替える為のスイッチであり、ハーフイタレーションに対して奇数回目か偶数回目かによって切り替え信号1405により制御されている。

【0116】インターリーブを伴わない処理においては、選択スイッチ1403はアップカウンタ1401を選択し、選択スイッチ1404はパリティビット1を選択する。

【0117】従って、情報系列が蓄積されているデュアルポートRAM1406の読み出しアドレスにアップカウンタ1401が接続されることになるので、デュアルポートRAM1406からはインターリーピングのされていない情報系列が出力される。同時にデュアルポートRAM1407の読み込みアドレスにもスイッチ1403を介してアップカウンタ1401が接続されることになるので、デュアルポートRAM1407からもインターリーピングのされていないアプリオリが出力される。

【0118】この二つの信号は加算器1408によって加え合わされ軟入力軟出力復号器(SISO)1410に入力される。SISO1410は、MAP復号を対数上で行う所謂LogMAPあるいはMax-LogMAPで構成されており、加算器1408による加算処理は確率演算における乗算に相当する。

【0119】この加算器1408の演算結果とスイッチ1404にて選択されたパリティビット1によってMAP演算が実行され、その結果から遅延器1411でタイミングを合わせた加算値が加算器1412で減算され次のアプリオリとしてデュアルポートRAM1407に入力される。デュアルポートRAM1407の書き込み用アドレスには遅延器1409を介して読み出し用アドレスと同じものがタイミングを合わせて入力されているので情報系列のシンボル位置に対応したアドレスにアプリオリデータが蓄積されることになる。

【0120】次に、ハーフイタレーションのインターリーブを伴う処理になると、選択スイッチ1403はインターリーピング順序発生器1402を選択し、選択スイッチ1404はパリティビット2を選択する。

【0121】従って、情報系列が蓄積されているデュアルポートRAM1406の読み出しアドレスにインターリーピング順序発生器1402が接続されることになるので、デュアルポートRAM1406からはインターリーピングがなされた情報系列が出力される。同時にデュアルポートRAM1407の読み込みアドレスにもスイッチ1403を介してインターリーピング順序発生器1402が接続されることになるので、デュアルポートRAM1407からもインターリーピングがなされたアプリオリが出力される。

【0122】この二つの信号は加算器1408によって加え合わされ軟入力軟出力復号器(SISO)1410に入力される。加算器1408による加算処理は確率演算における乗算に相当する。

【0123】この加算器1408の演算結果とスイッチ1404にて選択されたパリティビット2によってMAP演算が実行され、その結果から遅延器1411でタイミングを合わせた加算値が加算器1412で減算され次のアプリオリとしてデュアルポートRAM1407に入力される。デュアルポートRAM1407の書き込み用アドレスには遅延器1409を介して読み出し用アド

レスと同じものがタイミングを合わせて入力されているので元のアドレス位置にアプリアリデータが蓄積されることになる。即ちデインターリーブが施されたことになる。

【0124】このイタレーション処理によって復号性能を飛躍的に向上させるのがターボデコードの特徴であり、最終的に判定器1413により硬判定がなされ、出力端1414より復号データが出力される。

【0125】

【発明の効果】本発明によれば、各種のマルチメディアサービスにおいて、多種類のインターリーブングパターンを用意する必要がある場合であっても、そのために膨大なメモリ容量を必要とすることなく対応することができる。

【0126】また、例えば8イタレーション等の構成で2Mbps以上の受信データ系列の復号を行う場合であっても、生成されたインターリーブングパターンを一度高速メモリに蓄える為の高速なメモリ容量を必要とすることなく、更に、生成されたパターンを実際に処理を行っているターボデコード内のインターリーブング用RAMに転送する必要がある場合であっても、そのインターフェースがボトルネックとなる様な大量の転送データ量を必要とすることなく対応することができる。

【0127】更に、可変レート機能を持たせた場合、頻繁にインターリーブ長の変更が生じるが、そのような場合であっても、インターフェース上のボトルネックを助長する様なことがない最小限のパラメータ転送で実現出来、マルチメディアサービスにおける転送レートに追従出来ないといった問題を解消したインターリーブング順序発生器、インターリーバ、ターボエンコーダ及びターボデコードを提供することが出来る。

【図面の簡単な説明】

【図1】本発明のインターリーブング順序発生器の内、 $(\nu^{q_0})^j \pmod p \sim (\nu^{q_{R-1}})^j \pmod p$ の生成ブロックを示した図である。

【図2】本発明のインターリーブング順序発生器の内、行位置の加算を行ってインターリーブング順序を生成するブロック図である。

【図3】列数 $C = p - 1$ の場合であって、減算により入れ替えパターンを $0 \sim C - 1$ とした場合のインターリーブング順序を生成するブロック図である。

【図4】有限体上の高速乗算器を二つ用意してスキップしても淀み無く $(\nu^{q_0})^j \pmod p \sim (\nu^{q_{R-1}})^j \pmod p$ を生成するブロック図である。

【図5】スキップしても淀み無くインターリーブング順序を発生するブロックの内、行位置の加算を行ってインターリーブング順序を生成するブロック図である。

【図6】インターリーブング順序を表す信号に対し、インターリーバ対象範囲を超える信号をスキップするブロック図である。

【図7】有限体上の高速乗算器を示す図である。

【図8】モジュロ演算の構成要素を示す図である。

【図9】モジュロ演算の動作を説明した図である。

【図10】読み出しによるインターリーブング処理を表した図である。

【図11】書き込みによるデインターリーブング処理を表した図である。

【図12】デュアルポートRAMから読み出すことによるインターリーブング処理と書き込みによるデインターリーブングを同時に行った図である。

【図13】本発明のインターリーブング順序発生器を用いたターボエンコーダを示す図である。

【図14】本発明のインターリーブング順序発生器を用いたターボデコードを示す図である。

【図15】従来のターボエンコーダの構成例を示す図である。

【図16】従来のターボデコードの構成例を示す図である。

【図17】RAMに蓄積されたインターリーブング順序によりビット単位で並び替えを行う従来のターボデコードの構成例を示す図である。

【符号の説明】

101、102 レジスタ

103 有限体上の高速乗算器

104、105、106 セレクタ

107 出力

201 $(\nu^{q_0})^j \pmod p \sim (\nu^{q_{R-1}})^j \pmod p$ を生成するブロック

202 零の値を持った定数ブロック

203 pの値を持った定数ブロック

204 セレクタ

205 ブロック入れ替えパターンT(i)

206 二次元配列の列数を設定するブロック

207 乗算器

208 加算器

209、 インターリーブング順序出力

301 $(\nu^{q_0})^j \pmod p \sim (\nu^{q_{R-1}})^j \pmod p$ を生成するブロック

305 ブロック入れ替えパターンT(i)

306 二次元配列の列数p-1をもった定数ブロック

307 乗算器

308 加算器

309 インターリーブング順序出力

401、402 偶数用レジスタ

403 偶数用乗算器

404、405、406 セレクタ

407 出力

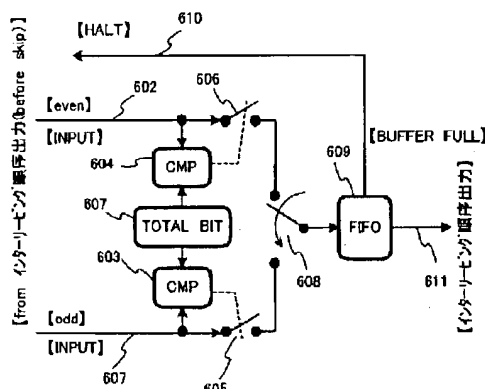
411、412 奇数用レジスタ

413 奇数用乗算器

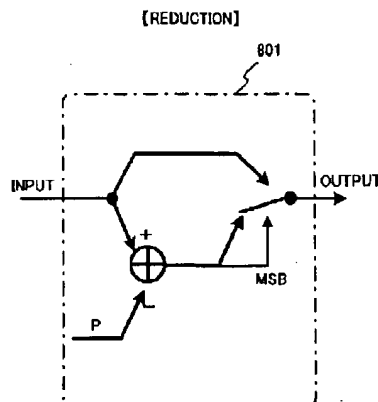
414、415、416 セクタ
 417 出力
 501 $(\ell^{\wedge}q_0)^{\wedge}j, (\ell^{\wedge}q_2)^{\wedge}j, \dots (\text{mod } p)$ を生成するブロック
 502 零の値を持った定数ブロック
 503 pの値を持った定数ブロック
 504 セクタ
 505 ブロック入れ替えパターンT(i)
 506 二次元配列の列数を設定するブロック
 507 乗算器
 508 加算器
 509 偶数に対応するインターリーブ順序出力
 511 $(\ell^{\wedge}q_1)^{\wedge}j, (\ell^{\wedge}q_3)^{\wedge}j, \dots (\text{mod } p)$ を生成するブロック
 512 零の値を持った定数ブロック
 513 pの値を持った定数ブロック
 514 セクタ
 517 乗算器
 518 加算器
 519 奇数に対応するインターリーブ順序出力
 601 奇数に対応するインターリーブ順序入力
 602 偶数に対応するインターリーブ順序入力
 603、604 コンパレータ
 605、606 スイッチ
 607 トータルビット数を値として持った定数ブロック
 608 切り替えスイッチ
 609 FIFO
 610 ホルト (HALT) 信号
 611 スキップ後のインターリーブ順序出力
 701 乗算器
 702 モジュロ演算器
 801 比較減算回路
 901 上位8ビットで比較減算される様子の説明
 1001 インターリーブ順序発生器

1002 RAM
 1101 インターリーブ順序発生器
 1102 RAM
 1201 インターリーブ順序発生器
 1202 デュアルポートRAM
 1203 遅延器
 1301 インターリーブ順序発生器
 1302 アップカウンタ
 1303 デュアルポートRAM
 10 1304 コンポーネントエンコーダ1
 1305 コンポーネントエンコーダ2
 1401 アップカウンタ
 1402 インターリーブ順序発生器
 1403、1404 選択スイッチ
 1405 切り替え信号
 1406、1407 デュアルポートRAM
 1408 加算器
 1409 遅延器
 1410 軟入力軟出力復号器 (SISO)
 1411 遅延器
 1412 加算器 (減算器)
 1413 判定器
 1414 復号データ出力
 1501 インターリーブ
 1502、1503 コンポーネントエンコーダ
 1601、1602 インターリーブ
 1603、1604 軟入力軟出力デコーダ (復号器) (SISO)
 1605 分離器
 1606、1607 デインターリーブ
 1608 判定器
 1701 インターリーブを行うデータ系列
 1702 インターリーブ順序が蓄積されたRAM
 1703 インターリーブ後のデータ系列

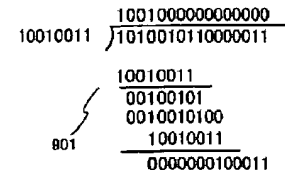
【図6】



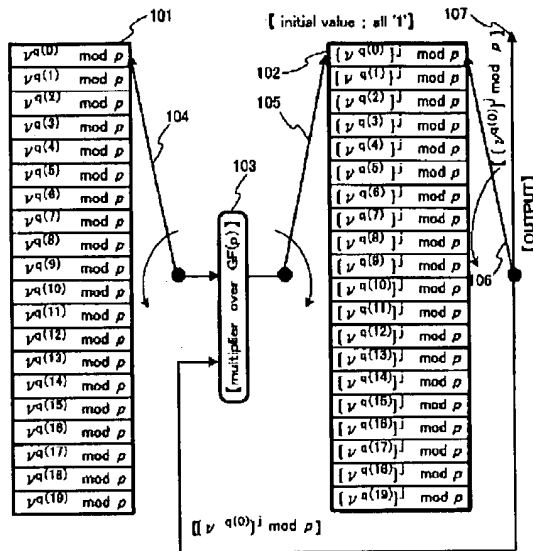
【図8】



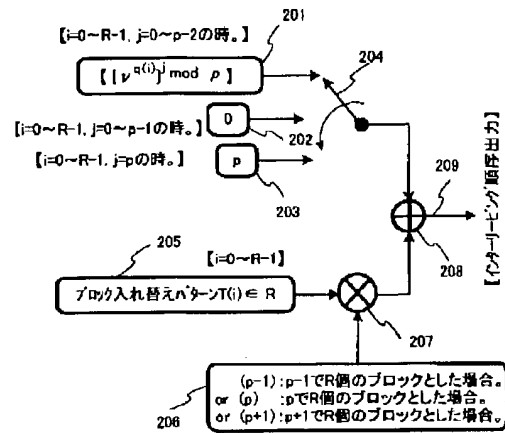
【図9】



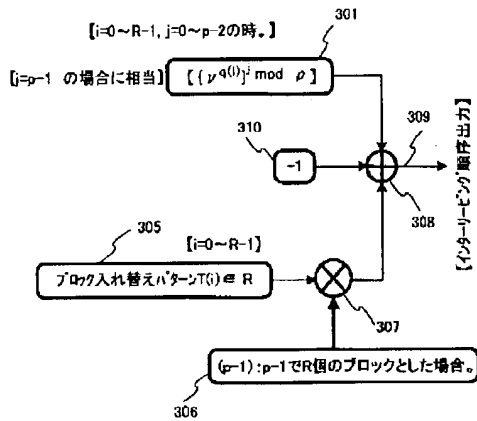
【図1】



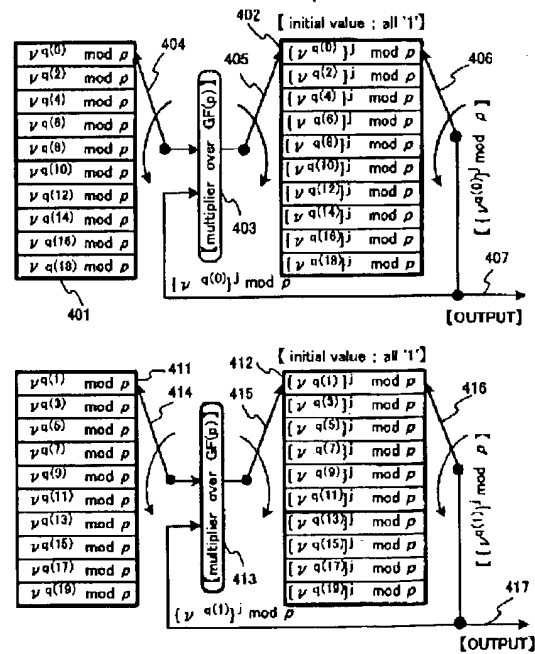
【図2】



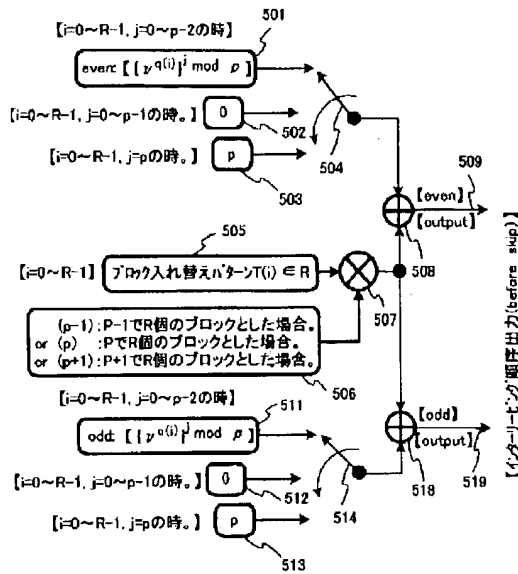
【図3】



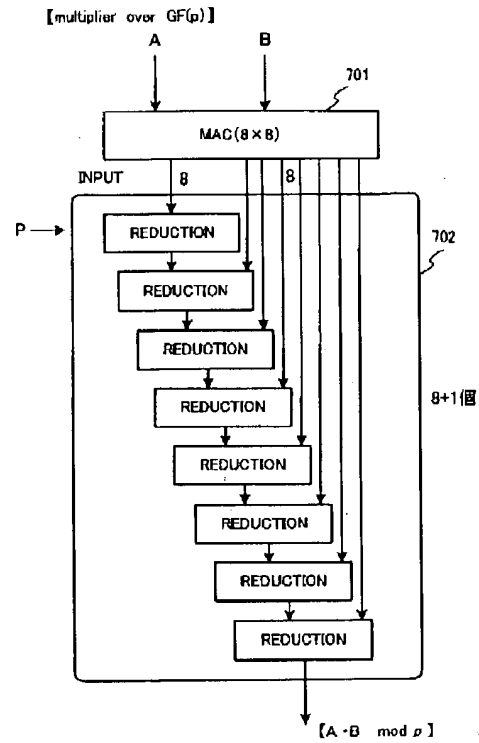
【図4】



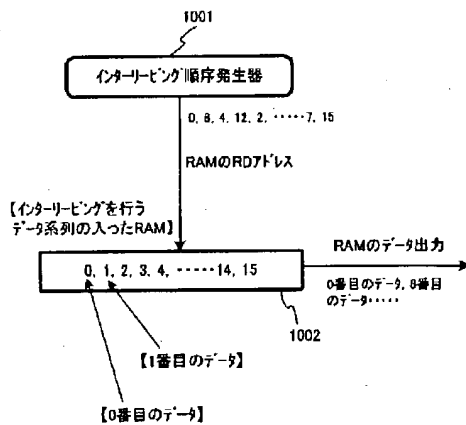
【図5】



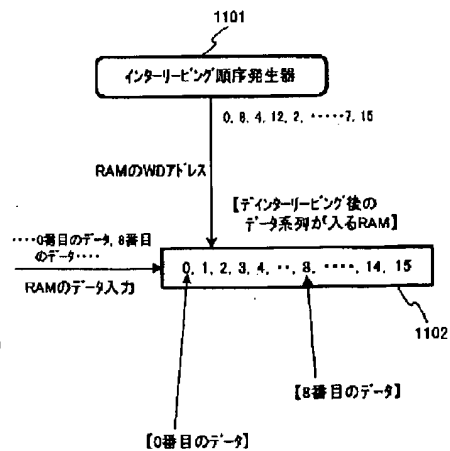
【図7】



【図10】



【図11】



【図17】

